A comparison between Markov approximations and other methods for large spatial data sets[☆]

David Bolin[a,*], Finn Lindgren[b]

[a] *Mathematical Statistics, Centre for Mathematical Sciences, Lund University, Box 118, SE-22100 Lund, Sweden*
[b] *Department of Mathematical Sciences, University of Bath, BA2 7AY, UK*

Abstract

The Matérn covariance function is a popular choice for modeling dependence in spatial environmental data. Standard Matérn covariance models are, however, often computationally infeasible for large data sets. Recent results for Markov approximations of Gaussian Matérn fields based on Hilbert space approximations are extended using wavelet basis functions. Using a simulation-based study, these Markov approximations are compared with two of the most popular methods for computationally efficient model approximations; covariance tapering and the process convolution method. The methods are compared with respect to their computational properties when used for spatial prediction (kriging), and the results show that, for a given computational cost, the Markov methods have a substantial gain in accuracy compared with the other methods.

Keywords: Matérn covariances; kriging; wavelets; Markov random fields; covariance tapering; process convolutions

1. Introduction

The traditional methods in spatial statistics were typically developed without any considerations of computational efficiency. In many of the classical applications of spatial statistics in environmental sciences, the cost for

[☆] Matlab programs for the comparisons can be obtained from the supplementary material of the electronic version of the paper.

[*] Corresponding author. Tel.: +46 46 2227974; fax: +46 46 2224623
Email address: bolin@maths.lth.se (David Bolin)

obtaining measurements limited the size of the data sets to ranges where computational cost was not an issue. Today, however, with the increasing use of remote sensing satellites, producing many large climate data sets, computational efficiency is often a crucial property.

In recent decades, several techniques for building computationally efficient models have been suggested. In many of these techniques, the main assumption is that a latent, zero-mean Gaussian process $X(\mathbf{s})$, $\mathbf{s} \in \mathbb{R}^d$, can be expressed, or at least approximated, through some finite basis expansion

$$X(\mathbf{s}) = \sum_{j=1}^{n} w_j \xi_j(\mathbf{s}), \tag{1}$$

where w_j are Gaussian random variables and $\{\xi_j\}_{j=1}^n$ are some pre-defined basis functions. The justification for using these basis expansions is usually that they converge to the true spatial model as n tends to infinity. However, for a finite n, the choice of the weights and basis functions will greatly affect the approximation error and the computational efficiency of the model. Hence, if one wants an accurate model for a given computational cost, asymptotic arguments are insufficient.

If the process $X(\mathbf{s})$ has a discrete spectral density, one can obtain an approximation of the form (1) by truncating the spectral expansion of the process. Another way to obtain an, in some sense optimal, expansion of the form (1) is to use the eigenfunctions of the covariance function for the latent field $X(\mathbf{s})$ as a basis, which is usually called the Karhunen-Loève (KL) transform (see e.g. Gelfand et al., 2010, Chapter 8). The problem with the KL transform is that analytic expressions for the eigenfunctions are only known in a few simple cases, which are often insufficient to represent the covariance structure in real data sets. Numerical approximations of the eigenfunctions can be obtained for a given covariance function; however, the covariance function is in most cases not known, but has to be estimated from the data. In these cases, it is infeasible to use the KL expansion in the parameter estimation, which is often the most computationally demanding part of the analysis. The spectral representation has a similar problem since the computationally efficient methods are usually restricted to stationary models with gridded data, and are not applicable in more general situations. Thus, to be useful for a broad range of practical applications, the methods should be applicable to a wide family of stationary covariance functions, and be extendable to nonstationary covariance structures.

One method that fulfills these requirements is the process convolution approach (Barry and Ver Hoef, 1996; Higdon, 2001; Cressie and Ravlicová, 2002; Rodrigues and Diggle, 2010). In this method, the stochastic field, $X(\mathbf{s})$, is defined as the convolution of a Gaussian white noise process with some convolution kernel $K(\mathbf{s})$. This convolution is then approximated by a sum of the form (1) to get a discrete model representation. Process convolution approximations are computationally efficient if a small number of basis functions can be used, but in practice, this will often give a poor approximation of the continuous convolution model.

A popular method for creating computationally efficient approximations is covariance tapering (Furrer et al., 2006). This method can not be written as an approximation of the form (1), but the idea is instead to taper the true covariance to zero beyond a certain range by multiplying the covariance function with some compactly supported taper function (Gneiting, 2002). This facilitates the use of sparse matrix techniques that increases the computational efficiency, at the cost of replacing the original model with a different model, which can lead to problems depending on the spatial structure of the data locations. However, the method is applicable to both stationary and nonstationary covariance models, and instead of choosing the set of basis functions in (1), the taper range and the taper function have to be chosen.

Nychka et al. (2002) used a wavelet basis in the expansion (1), and showed that by allowing for some correlation among the random variables w_j, one obtains a flexible model that can be used for estimating nonstationary covariance structures. As a motivating example, they showed that using a wavelet basis, computationally efficient approximations to the popular Matérn covariance functions can be obtained using only a few nonzero correlations for the weights w_j. The approximations were, however, obtained numerically, and no explicit representations were derived.

Rue and Tjelmeland (2002) showed that general stationary covariance models can be closely approximated by Markov random fields, by numerically minimizing the error in the resulting covariances. Song et al. (2008) extended the method by applying different loss criteria, such as minimizing the spectral error or the Kullback-Leibler divergence. A drawback of the methods is that, as for the KL and wavelet approaches, the numerical optimisation must in general be performed for each distinct parameter configuration.

Recently, Lindgren and Rue (2007) derived an explicit method for producing computationally efficient approximations to the Matérn covariance family. The method uses the fact that a random process on $\mathbb{R}^d$ with a Matérn

covariance function is a solution to a certain stochastic partial differential equation (SPDE). By considering weak solutions to this SPDE with respect to some set of local basis functions $\{\xi_j\}_{j=1}^n$, an approximation of the form (1) is obtained, where the stochastic weights have a sparse precision matrix (inverse covariance matrix), that can be written directly as a function of the parameters, without any need for costly numerical calculations. The method is also extendable to more general stationary and nonstationary models by extending the generating SPDE (Lindgren et al., 2011; Bolin and Lindgren, 2011).

In this paper, we use methods from Lindgren and Rue (2007) and Lindgren et al. (2011) to algebraically compute the weights w_j for wavelet-based approximations to Gaussian Matérn fields (Section 3). For certain wavelet bases, the weights form a Gaussian Markov Random Field (GMRF), which greatly increases the computational efficiency of the approximation. For other wavelet bases, such as the one used in Nychka et al. (2002), the weights can be well approximated with a GMRF.

In order to evaluate the practical usefulness of the different approaches, a detailed analysis of the computational aspects of the spatial prediction problem is performed (Section 2 and Section 4). The results show that the GMRF methods are more efficient and accurate than both the process convolution approach and the covariance tapering method for situations when the GMRF method is applicable.

2. Spatial prediction and computational cost

As a motivating example why computational efficiency is important, let us consider spatial prediction. The most widely used method for spatial prediction is commonly known as kriging in geostatistics. Let $Y(\mathbf{s})$ be an observation of a latent Gaussian field, $X(\mathbf{s})$, under zero-mean Gaussian measurement noise, $\mathcal{E}(\mathbf{s})$, uncorrelated with X and with covariance function $r_{\mathcal{E}}(\mathbf{s}, \mathbf{t})$,

$$Y(\mathbf{s}) = X(\mathbf{s}) + \mathcal{E}(\mathbf{s}), \tag{2}$$

and let $\mu(\mathbf{s})$ and $r(\mathbf{s}, \mathbf{t})$ be the mean value function and covariance function for $X(\mathbf{s})$ respectively. Depending on the assumptions on $\mu(\mathbf{s})$, kriging is usually divided into simple kriging (if μ is known), ordinary kriging (if μ is unknown but independent of $\mathbf{s}$), and universal kriging (if μ is unknown and can be expressed as a linear combination of some deterministic basis functions). To

limit the scope of this article, parameter estimation will not be considered, and to simplify the notations, we let $\mu(\mathbf{s}) \equiv 0$. It should, however, be noted that all results in later sections regarding computational efficiency also hold in the cases of ordinary kriging and universal kriging. For more details on kriging, see e.g. Stein (1999), Chiles and Delfiner (1999), or Schabenberger and Gotway (2005).

Let $r(\mathbf{s}, \mathbf{t})$ have some parametric structure, and let the vector $\boldsymbol{\gamma}$ contain all covariance parameters. Let $\mathbf{Y}$ be a vector containing the observations, $\mathbf{X}_1$ be a vector containing $X(\mathbf{s})$ evaluated at the measurement locations, $\mathbf{s}_1, \ldots, \mathbf{s}_m$, and let $\mathbf{X}_2$ be a vector containing $X(\mathbf{s})$ at the locations, $\mathbf{t}_1, \ldots, \mathbf{t}_{\hat{m}}$, for which the kriging predictor should be calculated. With $\mathbf{X} = (\mathbf{X}_1^\top, \mathbf{X}_2^\top)^\top$, one has $\mathbf{X}_1 = \mathbf{A}_1\mathbf{X}$, and $\mathbf{X}_2 = \mathbf{A}_2\mathbf{X}$ for two diagonal matrices $\mathbf{A}_1$ and $\mathbf{A}_2$, and the model can now be written as $\mathbf{X}|\boldsymbol{\gamma} \sim \mathsf{N}(\mathbf{0}, \boldsymbol{\Sigma}_X)$, and $\mathbf{Y}|\mathbf{X}, \boldsymbol{\gamma} \sim \mathsf{N}(\mathbf{A}_1\mathbf{X}, \boldsymbol{\Sigma}_\mathcal{E})$, where $\boldsymbol{\Sigma}_X$ is the covariance matrix for $\mathbf{X}$ and $\boldsymbol{\Sigma}_\mathcal{E}$ contains the covariances $r_\mathcal{E}(\mathbf{s}_i, \mathbf{s}_j)$. It is straightforward to show that the posterior is $\mathbf{X}|\mathbf{Y}, \boldsymbol{\gamma} \sim \mathsf{N}(\hat{\boldsymbol{\Sigma}}\mathbf{A}_1\boldsymbol{\Sigma}_\mathcal{E}^{-1}\mathbf{Y}, \hat{\boldsymbol{\Sigma}})$, where $\hat{\boldsymbol{\Sigma}} = (\boldsymbol{\Sigma}_X^{-1} + \mathbf{A}_1^\top\boldsymbol{\Sigma}_\mathcal{E}^{-1}\mathbf{A}_1)^{-1}$, and the well-known expression for the kriging predictor is given by the conditional mean

$$\begin{aligned}\mathsf{E}(\mathbf{X}_2|\mathbf{Y}, \boldsymbol{\gamma}) &= \mathbf{A}_2\hat{\boldsymbol{\Sigma}}\mathbf{A}_1\boldsymbol{\Sigma}_\mathcal{E}^{-1}\mathbf{Y} = \mathbf{A}_2\boldsymbol{\Sigma}_X\mathbf{A}_1^\top(\mathbf{A}_1\boldsymbol{\Sigma}_X\mathbf{A}_1^\top + \boldsymbol{\Sigma}_\mathcal{E})^{-1}\mathbf{Y} \\ &= \boldsymbol{\Sigma}_{X_2X_1}(\boldsymbol{\Sigma}_{X_1} + \boldsymbol{\Sigma}_\mathcal{E})^{-1}\mathbf{Y} = \boldsymbol{\Sigma}_{X_2X_1}\boldsymbol{\Sigma}_Y^{-1}\mathbf{Y}, \end{aligned} \tag{3}$$

where the elements on row i and column j in $\boldsymbol{\Sigma}_{X_2X_1}$ and $\boldsymbol{\Sigma}_Y$ are given by the covariances $r(\mathbf{t}_i, \mathbf{s}_j)$ and $r(\mathbf{s}_i, \mathbf{s}_j) + r_\mathcal{E}(\mathbf{s}_i, \mathbf{s}_j)$ respectively. To get the standard expression for the variance of the kriging predictor, the Woodbury identity is used on $\hat{\boldsymbol{\Sigma}}$:

$$\begin{aligned}\mathsf{V}(\mathbf{X}_2|\mathbf{Y}, \boldsymbol{\gamma}) &= \mathbf{A}_2(\boldsymbol{\Sigma}_X^{-1} + \mathbf{A}_1^\top\boldsymbol{\Sigma}_\mathcal{E}^{-1}\mathbf{A}_1)^{-1}\mathbf{A}_2^\top \\ &= \mathbf{A}_2\boldsymbol{\Sigma}_X\mathbf{A}_2 - \mathbf{A}_2\boldsymbol{\Sigma}_X\mathbf{A}_1^\top(\mathbf{A}_1\boldsymbol{\Sigma}_X\mathbf{A}_1^\top + \boldsymbol{\Sigma}_\mathcal{E})\mathbf{A}_1\boldsymbol{\Sigma}_X\mathbf{A}_2^\top \\ &= \boldsymbol{\Sigma}_{X_2} - \boldsymbol{\Sigma}_{X_2X_1}\boldsymbol{\Sigma}_Y^{-1}\boldsymbol{\Sigma}_{X_2X_1}^\top. \end{aligned}$$

If there are no simplifying assumptions on $\boldsymbol{\Sigma}_X$, the computational cost for calculating the kriging predictor is $\mathcal{O}(\hat{m}m + m^3)$, and the cost for calculating the variance is $\mathcal{O}(m^3 + m^2\hat{m} + \hat{m}^2m + \hat{m}^2)$ if standard methods for matrix mutiplication are used. This means that with 1000 measurements, the number of operations needed for the kriging prediction for a single location is on the order of 10^9. These computations are thus not feasible for a large data set where one might have more than 10^6 measurements.

The methods described in Section 1 all make different approximations in order to reduce the computational cost for calculating the kriging predictor and its variance. These different approximations, and their impact on the computational cost, are described in more detail in Section 4; however, to get a general idea of how the computational efficiency can be increased, consider the kriging predictor for a model of the form (1). The field $\mathbf{X}$ can then be written as $\mathbf{X} = \mathbf{B}\mathbf{w} \sim \mathsf{N}(\mathbf{0}, \mathbf{B}\boldsymbol{\Sigma}_w\mathbf{B}^\top)$, where column i in the matrix $\mathbf{B}$ contains the basis function $\xi_i(\mathbf{s})$ evaluated at all measurement locations and all locations where the kriging prediction is to be calculated and $\mathbf{w} = (w_1, \ldots, w_n)^\top$. Let $\mathbf{B}_1 = \mathbf{A}_1\mathbf{B}$ and $\mathbf{B}_2 = \mathbf{A}_2\mathbf{B}$ be the matrices containing the basis functions evaluated at the measurement locations and the kriging locations respectively. The kriging predictor is then

$$\mathsf{E}(\mathbf{X}_2|\mathbf{Y}, \boldsymbol{\gamma}) = \mathbf{B}_2(\boldsymbol{\Sigma}_w^{-1} + \mathbf{B}_1^\top\boldsymbol{\Sigma}_\mathcal{E}^{-1}\mathbf{B}_1)^{-1}\mathbf{B}_1\boldsymbol{\Sigma}_\mathcal{E}^{-1}\mathbf{Y}. \tag{4}$$

If $\mathcal{E}$ is Gaussian white noise, $\boldsymbol{\Sigma}_\mathcal{E}$ is diagonal and easy to invert. If $\boldsymbol{\Sigma}_w^{-1}$ is either known, or easy to calculate, the most expensive calculation in (4) is to solve $\mathbf{u} = (\boldsymbol{\Sigma}_w^{-1} + \mathbf{B}_1^\top\boldsymbol{\Sigma}_\mathcal{E}^{-1}\mathbf{B}_1)^{-1}\mathbf{B}_1\boldsymbol{\Sigma}_\mathcal{E}^{-1}\mathbf{Y}$. This is a linear system of n equations, where n is the number of basis functions used in the approximation. Thus, the easiest way of reducing the computational cost is to choose $n \ll m$, which is what is done in the convolution approach. Another approach is to ensure that $(\boldsymbol{\Sigma}_w^{-1} + \mathbf{B}_1^\top\boldsymbol{\Sigma}_\mathcal{E}^{-1}\mathbf{B}_1)$ is a sparse matrix. Sparse matrix techniques can then be used to calculate the kriging predictor, and the computational cost can be reduced without reducing the number of basis functions in the approximation. If a wavelet basis is used, $\mathbf{B}_1^\top\boldsymbol{\Sigma}_\mathcal{E}^{-1}\mathbf{B}_1$ is sparse, and in Section 3, it is shown that the precision matrix $\mathbf{Q}_w = \boldsymbol{\Sigma}_w^{-1}$ can also be chosen as a sparse matrix by using the Hilbert space approximation technique by Lindgren et al. (2011).

3. Wavelet approximations

In the remainder of this paper, the focus is on the family of Matérn covariance functions (Matérn, 1960) and the computational efficiency of some different techniques for approximating Gaussian Matérn fields. This section shows how wavelet bases can be used in the Hilbert space approximation technique by Lindgren et al. (2011) to obtain computationally efficient Matérn approximations.

3.1. The Matérn covariance family

Because of its versatility, the Matérn covariance family is one of the most popular choices for modeling spatial data. There are a few different paramet-

erizations of the Matérn covariance function in the literature, and the most suitable in our context is

$$r(\boldsymbol{\tau}) = \frac{2^{1-\nu}\phi^2}{(4\pi)^{\frac{d}{2}}\Gamma(\nu+\frac{d}{2})\kappa^{2\nu}}(\kappa\|\boldsymbol{\tau}\|)^\nu K_\nu(\kappa\|\boldsymbol{\tau}\|), \quad \boldsymbol{\tau}\in\mathbb{R}^d, \tag{5}$$

where $\nu > 0$ is a shape parameter, κ^2 a scale parameter, ϕ^2 a variance parameter, and K_ν is a modified Bessel function of the second kind of order $\nu > 0$. The reason for using this slightly non-standard parameterization is that it reflects the solution to the SPDE defined in (7). With this parametrization, the variance is $r(\mathbf{0}) = \phi^2\Gamma(\nu)(4\pi)^{-\frac{d}{2}}\Gamma(\nu+\frac{d}{2})^{-1}\kappa^{-2\nu}$, and the associated spectral density is

$$S(\boldsymbol{\omega}) = \frac{\phi^2}{(2\pi)^d}\frac{1}{(\kappa^2+\|\boldsymbol{\omega}\|^2)^{\nu+\frac{d}{2}}}. \tag{6}$$

For the special case $\nu = 0.5$, the Matérn covariance function is the exponential covariance function. The smoothness of the field increases with ν, and in the limit as $\nu\to\infty$, the covariance function is a Gaussian covariance function if κ is also scaled accordingly, which gives an infinitely differentiable field.

3.2. Hilbert space approximations

As noted by Whittle (1963), a random process with the covariance (5) is a solution to the SPDE

$$(\kappa^2-\Delta)^{\frac{\alpha}{2}}X(\mathbf{s}) = \phi\,\mathcal{W}(\mathbf{s}), \tag{7}$$

where $\mathcal{W}(\mathbf{s})$ is Gaussian white noise, Δ is the Laplacian, and $\alpha = \nu + d/2$. The key idea in Lindgren et al. (2011) is to approximate the solution to the SPDE using a basis expansion of the form (1). The starting point of the approximation is to consider the stochastic weak formulation of the SPDE

$$\left\{\left\langle b_i,\,(\kappa^2-\Delta)^{\frac{\alpha}{2}}X\right\rangle, i=1,\ldots,n_b\right\} \overset{d}{=} \left\{\left\langle b_i,\,\phi\,\mathcal{W}\right\rangle, i=1,\ldots,n_b\right\}. \tag{8}$$

Here $\overset{d}{=}$ denotes equality in distribution, $\langle f,\,g\rangle = \int f(\mathbf{s})g(\mathbf{s})\,\mathrm{d}\mathbf{s}$, and equality should hold for every finite set of test functions $\{b_i, i=1,\ldots,n_b\}$ from some appropriate space. A finite element approximation of the solution X is then obtained by representing it as a finite basis expansion of the form (1), where the stochastic weights are calculated by requiring (8) to hold for only a

specific set of test functions $\{b_i, i = 1, \ldots, n\}$. We illustrate the more general results from Lindgren et al. (2011) with the special case $\alpha = 2$, where one uses $b_i = \xi_i$ and then has

$$\left\langle \xi_i,\, (\kappa^2 - \Delta)X \right\rangle = \sum_{j=1}^{n} w_j \left\langle \xi_i,\, (\kappa^2 - \Delta)\xi_j \right\rangle . \tag{9}$$

By introducing the matrix $\mathbf{K}$ with elements $\mathbf{K}_{i,j} = \langle \xi_i,\, (\kappa^2 - \Delta)\xi_j \rangle$ and the vector $\mathbf{w} = (w_1, \ldots, w_n)^\top$, the left hand side of (8) can be written as $\mathbf{Kw}$. Since, by Lemma 1 in Lindgren et al. (2011)

$$\left\langle \xi_i,\, (\kappa^2 - \Delta)\xi_j \right\rangle = \kappa^2 \left\langle \xi_i,\, \xi_j \right\rangle - \left\langle \xi_i,\, \Delta\xi_j \right\rangle = \kappa^2 \left\langle \xi_i,\, \xi_j \right\rangle + \left\langle \nabla\xi_i,\, \nabla\xi_j \right\rangle ,$$

the matrix $\mathbf{K}$ can be written as the sum $\mathbf{K} = \kappa^2\mathbf{C} + \mathbf{G}$ where $\mathbf{C}_{i,j} = \langle \xi_i,\, \xi_j \rangle$ and $\mathbf{G}_{i,j} = \langle \nabla\xi_i,\, \nabla\xi_j \rangle$. The right hand side of (8) can be shown to be Gaussian with mean zero and covariance $\phi^2\mathbf{C}$ and one thus have that $\mathbf{w} \sim \mathsf{N}(\mathbf{0}, \phi^2\mathbf{K}^{-1}\mathbf{C}\mathbf{K}^{-1})$.

For the second fundamental case, $\alpha = 1$, Lindgren et al. (2011) show that $\mathbf{w} \sim \mathsf{N}(\mathbf{0}, \phi^2\mathbf{K}^{-1})$ and for higher order $\alpha \in \mathbb{N}$, the weak solution is obtained recursively using these two fundamental cases. For example, if $\alpha = 4$ the solution to $(\kappa^2 - \Delta)^2 X_0(\mathbf{s}) = \phi\,\mathcal{W}(\mathbf{s})$ is obtained by solving $(\kappa^2 - \Delta)X_0(\mathbf{s}) = \tilde{X}(\mathbf{s})$, where $\tilde{X}$ is the solution for the case $\alpha = 2$. This results in a precision matrix for the weights $\mathbf{Q}_\alpha$ defined recursively as

$$\mathbf{Q}_\alpha = \mathbf{K}\mathbf{C}^{-1}\mathbf{Q}_{\alpha-2}\mathbf{C}^{-1}\mathbf{K}, \quad \alpha = 3, 4, \ldots \tag{10}$$

where $\mathbf{Q}_1 = \phi^{-2}\mathbf{K}$ and $\mathbf{Q}_2 = \phi^{-2}\mathbf{K}^\top\mathbf{C}^{-1}\mathbf{K}$. Thus, all Matérn fields with $\nu + d/2 \in \mathbb{N}$ can be approximated through this procedure. For more details, see Lindgren and Rue (2007) and Lindgren et al. (2011). The results from Rue and Tjelmeland (2002) show that accurate Markov approximations exist also for other ν-values, and one approximate approach to finding explicit expressions for such models was given in the authors' response in Lindgren et al. (2011). However, in many practical applications ν cannot be estimated reliably (Zhang, 2004), and using only a discrete set of ν-values is not a significant restriction for $\nu \geq 1$.

3.3. Wavelet basis functions

In the previous section, nothing was said about how the basis functions $\{\xi_i\}$ should be chosen. The following sections, however, show that wavelet

bases have many desirable properties which makes them suitable to use in the Hilbert space approximations on $\mathbb{R}^d$. In this section, a brief introduction to multiresolution analysis and wavelets is given.

A multiresolution analysis on $\mathbb{R}$ is a sequence of closed approximation subspaces $\{V_j\}_{j\in\mathbb{Z}}$ of functions in $L^2(\mathbb{R})$ such that $V_j \subset V_{j+1}$, $\mathrm{cl}\bigcup_{j\in\mathbb{Z}} V_j = L^2(\mathbb{R})$, and $\bigcap_{j\in\mathbb{Z}} V_j = \{0\}$, where cl is the closure, and $f(s) \in V_j$ if and only if $f(2^{-j}s) \in V_0$. This last requirement is the multiresolution requirement because this implies that all the approximation spaces V_j are scaled versions of the space V_0. A multiresolution analysis is generated starting with a function usually called a father function or a scaling function. The function $\varphi \in L^2(\mathbb{R})$ is called a scaling function for $\{V_j\}_{j\in\mathbb{Z}}$ if it satisfies the two-scale relation

$$\varphi(s) = \sum_{k\in\mathbb{Z}} p_k \varphi(2s-k), \tag{11}$$

for some square-summable sequence $\{p_k\}_{k\in\mathbb{Z}}$ and the translates $\{\varphi(s-k)\}_{k\in\mathbb{Z}}$ form an orthonormal basis for V_0. Given the multiresolution analysis $\{V_j\}_{j\in\mathbb{Z}}$, the wavelet spaces $\{W_j\}_{j\in\mathbb{Z}}$ are then defined as the orthogonal complements of V_j in V_{j+1} for each j, and one can show that W_j is the span of $\{\psi(2^j s - k)\}_{k\in\mathbb{Z}}$, where the wavelet ψ is defined as $\psi(s) = \sum_{k\in\mathbb{Z}} (-1)^k \overline{p_{1-k}} \varphi(2s-k)$.

Given the spaces W_j, V_j can be decomposed as the direct sum

$$V_j = V_0 \oplus W_0 \oplus W_1 \oplus \ldots \oplus W_{j-1}. \tag{12}$$

Several choices of scaling functions have been presented in the literature. Among the most widely used constructions are the B-spline wavelets (Chui and Wang, 1992) and the Daubechies wavelets (Daubechies, 1992) that both have several desirable properties for our purposes.

The scaling function of B-spline wavelets are mth order B-splines with knots at the integers. Because of this, there exists closed form expressions for the corresponding wavelets, and the wavelets have compact support since the mth order scaling function has support on $(0, m+1)$. The wavelets are orthogonal at different scales, but translates at the same scale are not orthogonal. This property is usually referred to as semi-orthogonality.

The Daubechies wavelets form a hierarchy of compactly supported orthogonal wavelets that are constructed to have the highest number of vanishing moments for a given support width. This generates a family of wavelets with an increasing degree of smoothness. Except for the first Daubechies wavelet, there are no closed form expressions for these wavelets; however, for practical

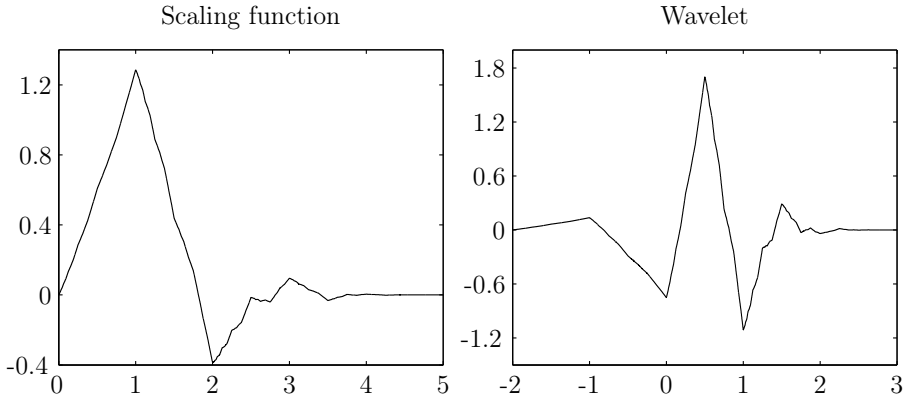

Figure 1: The DB3 scaling function and wavelet.

purposes, this is not a problem because the exact values for the wavelets at dyadic points can be obtained very fast using the Cascade algorithm (Burrus et al., 1988). In this work, the Daubechies 3 (DB3) wavelet is used because it is the first wavelet in the family that has one continuous derivative. The DB3 wavelet and its scaling function are shown in Figure 1.

3.4. Explicit wavelet Hilbert space approximations

To use the Hilbert space approximation for a given basis, the precision matrix, $\mathbf{Q}_\alpha$, for the weights has to be calculated. By (10), we only have to be able to calculate the matrices $\mathbf{C}$ and $\mathbf{G}$ to build the precision matrix for any $\alpha \in \mathbb{N}$. The elements in these matrices are inner products between the basis functions:

$$\mathbf{C}_{i,j} = \int \xi_i(\mathbf{s})\xi_j(\mathbf{s})\,\mathrm{d}\mathbf{s}, \text{ and} \qquad \mathbf{G}_{i,j} = \int (\nabla\xi_i(\mathbf{s}))^\top \nabla\xi_j(\mathbf{s})\,\mathrm{d}\mathbf{s}. \tag{13}$$

This section shows how these elements can be calculated for the DB3 wavelets and the B-spline wavelets. When using a wavelet basis in practice, one always has to choose a finest scale, J, to work with. Given that the subspace V_J is used as an approximation of $L^2(\mathbb{R})$, one can either use the direct basis for V_J, that consists of scaled and translated versions of the father function $\varphi(s)$, or one can use the multiresolution decomposition (12). In what follows, both cases are considered.

3.4.1. Daubechies wavelets on $\mathbb{R}$

Since the Daubechies wavelets form an orthonormal basis for $L^2(\mathbb{R})$, the matrix $\mathbf{C}$ is the identity matrix. Thus, only the matrix $\mathbf{G}$ has to be calculated in the case of Daubechies wavelets. If the direct basis for V_J is used, the matrix $\mathbf{G}$ contains inner products of the form

$$\left\langle \nabla\varphi(2^J s - k),\, \nabla\varphi(2^J s - l) \right\rangle = 2^J \left\langle \nabla\varphi(s),\, \nabla\varphi(s - l + k) \right\rangle \equiv 2^J \Lambda(k - l). \tag{14}$$

Because the scaling function has compact support on $[0, 2N-1]$, these inner products are non-zero only if $k - l \in [-(2N-2), 2N-2]$. Thus, the matrix $\mathbf{G}$ is sparse, which implies that the weights $\mathbf{w}$ in (1) form a GMRF. Since there are no closed form expressions for the Daubechies wavelets, there is no hope in finding a closed form expression for the non-zero inner products (14). Furthermore, standard numerical quadrature for calculating the inner products is too inaccurate due to the highly oscillating nature of the gradients. However, utilizing properties of the wavelets, one can calculate an approximation of the inner product of arbitrary precision by solving a system of linear equations as explained in Latto et al. (1991).

Using this technique for the DB3 wavelets, the following nonzero values for $\Lambda(\eta)$ are obtained: $\Lambda(0) = 5.267$, $\Lambda(\pm 1) = -3.390$, $\Lambda(\pm 2) = 0.876$, $\Lambda(\pm 3) = -0.114$, and $\Lambda(\pm 4) = -0.00535$. These values are calculated once and tabulated for constructing the $\mathbf{G}$ matrix, which is a band matrix with $2^J\Lambda(0)$ on the main diagonal, $2^J\Lambda(1)$ on the first off diagonals, et cetera.

If the multiresolution decomposition (12) is used as a basis for V_J, one also needs the inner products $\langle \nabla\psi(2^j s - k),\, \nabla\psi(2^i s - l)\rangle$, $i, j \in \mathbb{Z}$. Because of the two-scale relation (11), every wavelet $\psi(2^j s - k)$ can be written as a finite sum of translated scaling functions at scale J. Using this property, the $\mathbf{G}$ matrix can be constructed efficiently using the tabulated Λ values. Figure 2 shows the structure of the $\mathbf{G}$ matrices for a multiresolution DB3 basis with five layers of wavelets and the corresponding direct basis. Note that there are fewer non-zero elements in the precision matrix for the direct basis. Hence, it is more computationally efficient to use the direct basis instead of the multiresolution basis.

3.4.2. B-spline wavelets on $\mathbb{R}$

For the B-spline wavelets, the matrices $\mathbf{C}$ and $\mathbf{G}$ can be calculated directly using the closed form expressions for the basis functions and their derivatives. When a direct basis is used on $\mathbb{R}$, $\mathbf{C}$ is a band matrix with bandwidth $m+1$,

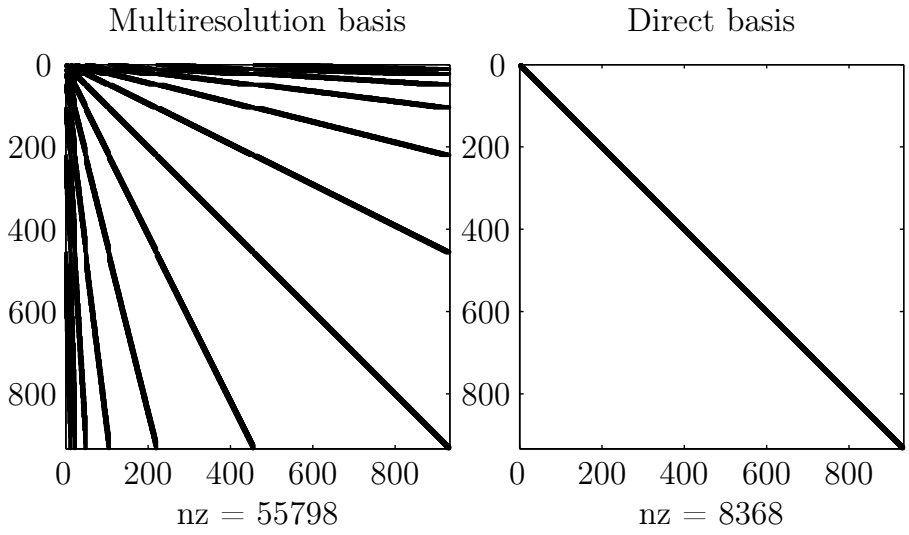

Figure 2: The non-zero elements in the $\mathbf{G}$ matrices for a multiresolution DB3 basis with five layer of wavelets and the corresponding direct basis. 6.4% of the elements are non-zero for the multiresolution basis whereas only 0.96% of the elements are non-zero for the direct basis.

if the mth order spline wavelet is used. For example, for $m = 1$, calculating (13) gives

$$\mathbf{C}_{i,j} = 2^{-J} \cdot \begin{cases} 2/3, & i = j, \\ 1/6, & |i-j| = 1, \\ 0 & \text{otherwise}, \end{cases} \quad \text{and} \quad \mathbf{G}_{i,j} = 2^{J} \cdot \begin{cases} 2, & i = j, \\ -1, & |i-j| = 1, \\ 0 & \text{otherwise}. \end{cases}$$

Since the expression for the precision matrix for the weights $\mathbf{w}$ contains the inverse of $\mathbf{C}$, it is a dense matrix. Hence, $\mathbf{C}^{-1}$ has to be approximated with a sparse matrix if $\mathbf{Q}$ should be sparse. This issue is addressed in Lindgren et al. (2011) by lowering the integration order of $\langle \xi_i, \xi_j \rangle$, which results in an approximate, diagonal $\mathbf{C}$ matrix, $\tilde{\mathbf{C}}$, with diagonal elements $\tilde{\mathbf{C}}_{i,i} = \sum_{k=1}^{n} \mathbf{C}_{i,k}$. In Section 4, the effect of this approximation on the covariance approximation for the basis expansion is studied in some detail. For the multiresolution basis, the matrices are block diagonal, and this approximation is not applicable.

3.4.3. Wavelets on $\mathbb{R}^d$

The easiest way of constructing a wavelet basis for $L^2(\mathbb{R}^d)$ is to use the tensor product functions generated by d one-dimensional wavelet bases. Let φ be the scaling function for a multiresolution on $\mathbb{R}$, the father function can be written as $\bar{\varphi}(x_1, \ldots, x_d) = \prod_{i=1}^d \varphi(x_i)$. The scalar product $\langle \nabla\bar{\varphi}(\mathbf{x}), \nabla\bar{\varphi}(\mathbf{x}+\boldsymbol{\eta})\rangle$, where $\boldsymbol{\eta} \in \mathbb{Z}^d$, can then be written as

$$\begin{aligned}
\langle \nabla\bar{\varphi}(\mathbf{x}), \nabla\bar{\varphi}(\mathbf{x}+\boldsymbol{\eta})\rangle &= \left\langle \nabla \prod_{i=1}^d \varphi(x_i), \nabla \prod_{i=1}^d \varphi(x_i+\eta_i) \right\rangle \\
&= \sum_{i=1}^d \int_{\mathbb{R}^d} \frac{\partial\,\varphi(x_i)}{\partial\, x_i} \frac{\partial\,\varphi(x_i+\eta_i)}{\partial\, x_i} \prod_{j\neq i} \varphi(x_j)\varphi(x_j+\eta_j)\,\mathrm{d}\mathbf{x} \\
&= \sum_{i=1}^d \Lambda(\eta_i) \prod_{j\neq i} \int_{\mathbb{R}} \varphi(x_j)\varphi(x_j+\eta_j)\,\mathrm{d}x_j.
\end{aligned}$$

This expression looks rather complicated but implies a simple Kronecker structure for $\mathbf{G}_d$, the $\mathbf{G}$ matrix in $\mathbb{R}^d$. For example, in $\mathbb{R}^2$ and $\mathbb{R}^3$,

$$\begin{aligned}
&\mathbf{G}_2 = \mathbf{G}_1 \otimes \mathbf{C}_1 + \mathbf{C}_1 \otimes \mathbf{G}_1, \text{ and} \\
&\mathbf{G}_3 = \mathbf{G}_1 \otimes \mathbf{C}_1 \otimes \mathbf{C}_1 + \mathbf{C}_1 \otimes \mathbf{G}_1 \otimes \mathbf{C}_1 + \mathbf{C}_1 \otimes \mathbf{C}_1 \otimes \mathbf{G}_1,
\end{aligned}$$

where $\mathbf{G}_1$ and $\mathbf{C}_1$ are the $\mathbf{G}$ and $\mathbf{C}$ matrices for the corresponding one-dimensional basis and $\otimes$ denotes the Kronecker product. Similarly, $\mathbf{C}_2 = \mathbf{C}_1 \otimes \mathbf{C}_1$, and $\mathbf{C}_3 = \mathbf{C}_1 \otimes \mathbf{C}_1 \otimes \mathbf{C}_1$. These expressions hold both if the direct basis for V_J is used or if the multiresolution construction (12) is used for the one-dimensional spaces. For Daubechies wavelets, the $\mathbf{C}$ matrix is the identity matrix for all $d \geq 1$. This also holds for the direct B-spline basis if the diagonal approximation is used for $\mathbf{C}_1$.

4. Comparison

As discussed in Section 2, computational efficiency is often an important aspect in practical applications. However, the computation time for obtaining, for example, an approximate kriging prediction is in itself not that interesting unless one also knows how accurate it is. We will, therefore, in this section compare the wavelet Markov approximations with two other popular methods, covariance tapering and process convolutions, with respect to their accuracy and computationally efficiency when used for kriging.

Before the comparison, we give a brief introduction to the process convolution method and the covariance tapering method and discuss the methods' computational properties. As mentioned in Section 2, the computational cost for the kriging prediction for a single location based on m observations is $\mathcal{O}(m^3)$. In what follows, the corresponding computational costs for the three different approximation methods are presented. We start with the wavelet Markov approximations and then look at the process convolutions and the covariance tapering method. After this, an initial comparison of the different wavelet approximations is performed in Section 4.4 and then the full kriging comparison is presented in Section 4.5-4.6.

4.1. Wavelet approximations

When using a wavelet basis, one can either work with the direct basis for the approximation space V_J or do the wavelet decomposition into the direct sum of $J-1$ wavelet spaces and V_0. If one only is interested in the approximation error, the decomposition into wavelet spaces is not necessary and it is more efficient to work in the direct basis for V_J since this will result in a precision matrix with fewer nonzero elements. We therefore only use the direct bases for V_J in the comparisons in this section.

The wavelet approximations are of the form (1), so Equation (4) is used to calculate the kriging predictor. However, since an explicit expression for the precision matrix for the weights $\mathbf{w}$ exists for this method, we rewrite the equation as

$$\mathsf{E}(\mathbf{X}_2|\mathbf{Y},\boldsymbol{\gamma}) = \mathbf{B}_2(\mathbf{Q}_w + \mathbf{B}_1^\top \mathbf{Q}_\mathcal{E}\mathbf{B}_1)^{-1}\mathbf{B}_1\mathbf{Q}_\mathcal{E}\mathbf{Y},$$

where $\mathbf{Q}_\mathcal{E} = \boldsymbol{\Sigma}_\mathcal{E}^{-1}$ is diagonal if $\mathcal{E}$ is Gaussian white noise. If the number of kriging locations is small, the computationally demanding step is to solve a system of the form

$$\mathbf{u} = (\mathbf{Q}_w + \mathbf{B}_1^\top \mathbf{Q}_\mathcal{E}\mathbf{B}_1)^{-1}\mathbf{v}.$$

Now, if the Daubechies wavelets or the Markov approximated spline wavelets are used, both $\mathbf{Q}_w$ and $\mathbf{B}_1^\top \mathbf{Q}_\mathcal{E}\mathbf{B}_1$ are sparse and positive definite matrices. The system is therefore most efficiently solved using Cholesky factorization, forward substitution, and back substitution. The forward substitution and back substitution are much faster than calculating the Cholesky triangle $\mathbf{L}$, so the computational complexity for the kriging predictor is determined by the calculation of $\mathbf{L}$. Because of the sparsity structure, this computational cost is in general $\mathcal{O}(n)$, $\mathcal{O}(n^{3/2})$, and $\mathcal{O}(n^2)$ for problems in one, two, and three

dimensions respectively (see Rue and Held, 2005). If the spline bases are used without the markov approximation, the computational cost is $O(n^3)$ since $\mathbf{Q}_w$ then is dense. It should be noted here that any basis could be used in the SPDE approximation, but in order to get good computational properties we need both $\mathbf{Q}_w$ and $\mathbf{B}_1^\top \mathbf{Q}_\mathcal{E} \mathbf{B}_1$ to be sparse. This is the reason for why Fourier bases, for instance, are not appropriate to use in the SPDE formulation since $\mathbf{B}_1$ in this case always is a dense matrix.

4.2. Process convolutions

In the process convolution method, the Gaussian random field $X(\mathbf{s})$ on $\mathbb{R}^d$ is specified as a process convolution

$$X(\mathbf{s}) = \int K(\mathbf{s}, \mathbf{u}) \mathcal{B}(\,\mathrm{d}\mathbf{u}), \tag{15}$$

where K is some deterministic kernel function and $\mathcal{B}$ is a Brownian sheet. One of the advantages with this construction is that nonstationary fields easily are constructed by allowing the convolution kernel to be dependent on location. If, however, the process is stationary one has $K(\mathbf{s}, \mathbf{u}) = K(\mathbf{s} - \mathbf{u})$ and the covariance function for X is $r(\boldsymbol{\tau}) = \int K(\mathbf{u} - \boldsymbol{\tau}) K(\mathbf{u}) \,\mathrm{d}\mathbf{u}$. Thus, the covariance function and the kernel K are related through

$$K = \mathcal{F}^{-1}\left(\frac{1}{(2\pi)^{\frac{d}{2}}} \sqrt{\mathcal{F}(r)} \right) = \mathcal{F}^{-1}\left(\frac{1}{(2\pi)^{\frac{d}{2}}} \sqrt{S} \right),$$

where S is the spectral density for $X(\mathbf{s})$ and $\mathcal{F}$ denotes the Fourier transform (Higdon, 2001). Since the spectral density for a Matérn covariance function in dimension d with parameters ν, ϕ^2, and κ is given by (6), one finds that the corresponding kernel is a Matérn covariance function with parameters $\nu_k = \frac{\nu}{2} - \frac{d}{4}$, $\phi_k^2 = \phi$, and $\kappa_k = \kappa$.

An approximation of (15) which is commonly used in convolution-based modeling is

$$X(\mathbf{s}) \approx \sum_{j=1}^{n} k(\mathbf{s} - \mathbf{u}_j) w_j,$$

where $\mathbf{u}_1, \ldots, \mathbf{u}_n$ are some fixed locations in the domain, and w_j are independent zero-mean Gaussian variables with variances equal to the area associated with each $\mathbf{u}_j$. Thus, this approximation is of the form (1), with basis functions $\xi_j(\mathbf{s}) = K(\mathbf{s} - \mathbf{u}_j)$. With this approximation, Equation (4)

is used to calculate the kriging predictor. Because the basis functions in the expansion are Matérn covariance functions, the matrices $\mathbf{B}_1$ and $\mathbf{B}_2$ are dense. Thus, even though both $\mathbf{\Sigma}_{\mathcal{E}}$ and $\mathbf{\Sigma}_w^{-1}$ are diagonal matrices, one has to solve a system of the form

$$\mathbf{u} = (\mathbf{\Sigma}_w^{-1} + \mathbf{B}_1^\top \mathbf{\Sigma}_{\mathcal{E}}^{-1} \mathbf{B}_1)^{-1} \mathbf{v},$$

where $(\mathbf{\Sigma}_w^{-1} + \mathbf{B}_1^\top \mathbf{\Sigma}_{\mathcal{E}}^{-1} \mathbf{B}_1)$ is a dense $n \times n$ matrix and n is the number of basis functions used in the basis expansion. The computational cost for both constructing and inverting the matrix is $\mathcal{O}(mn^2 + n^3)$. For kriging prediction of $\hat{m}$ locations, the total computational complexity is $\mathcal{O}(\hat{m}n + mn^2 + n^3)$.

4.3. Covariance tapering

Covariance tapering is not a method for constructing covariance models, but a method for approximating a given covariance model to increase the computational efficiency. The idea is to taper the true covariance, $r(\boldsymbol{\tau})$, to zero beyond a certain range, θ, by multiplying the covariance function with some compactly supported positive definite taper function $r_\theta(\boldsymbol{\tau})$. Using the tapered covariance, $r_{tap}(\boldsymbol{\tau}) = r_\theta(\boldsymbol{\tau}) r(\boldsymbol{\tau})$, the matrix $\mathbf{\Sigma}_Y$ in the expression for the kriging predictor (3) is sparse, which facilitates the use of sparse matrix techniques that increases the computational efficiency. The taper function should, of course, also be chosen such that the basic shape of the true covariance function is preserved, and of especial importance for asymptotic considerations is that the smoothness at the origin is preserved.

Furrer et al. (2006) studied the accuracy and numerical efficiency of tapered Matérn covariance functions, and suggested the following taper functions

$$\text{Wendland}_1\text{:} \quad r_\theta(\boldsymbol{\tau}) = \left(\max\left[1 - \frac{\|\boldsymbol{\tau}\|}{\theta}, 0 \right] \right)^4 \left(1 + 4\frac{\|\boldsymbol{\tau}\|}{\theta} \right),$$

$$\text{Wendland}_2\text{:} \quad r_\theta(\boldsymbol{\tau}) = \left(\max\left[1 - \frac{\|\boldsymbol{\tau}\|}{\theta}, 0 \right] \right)^6 \left(1 + 6\frac{\|\boldsymbol{\tau}\|}{\theta} + \frac{35\|\boldsymbol{\tau}\|^2}{2\theta^2} \right).$$

These taper functions were first introduced by Wendland (1995). For dimension $d \leq 3$, the Wendland_1 function is a valid taper function for the Matérn covariance function if $\nu \leq 1.5$, and the Wendland_2 function is a valid taper function if $\nu \leq 2.5$. Furrer et al. (2006) found that Wendland_1 was slightly better than Wendland_2 for a given ν, so we use Wendland_1 for all cases when $\nu \leq 1.5$ and Wendland_2 if $1.5 < \nu \leq 2.5$.

If a tapered Matérn covariance is used, the kriging predictor can be written as

$$\mathsf{E}(\mathbf{X}_2|\mathbf{Y},\boldsymbol{\gamma}) = \boldsymbol{\Sigma}^{tap}_{X_2X_1}(\boldsymbol{\Sigma}^{tap}_{X_1} + \boldsymbol{\Sigma}_{\mathcal{E}})^{-1}\mathbf{Y},$$

where the element on row i and column j in $\boldsymbol{\Sigma}^{tap}_{X_2X_1}$ and $\boldsymbol{\Sigma}^{tap}_{X_1}$ are given by $r_{tap}(\mathbf{t}_i, \mathbf{s}_j)$ and $r_{tap}(\mathbf{s}_i, \mathbf{s}_j)$ respectively. Since the tapered covariance is zero for lags larger than the taper range, θ, many of the elements in $\boldsymbol{\Sigma}^{tap}_{X_1}$ will be zero. Thus, the three step approach used for the wavelet Markov approximations can be used to solve the system $\mathbf{u} = (\boldsymbol{\Sigma}^{tap}_{X_1} + \boldsymbol{\Sigma}_{\mathcal{E}})^{-1}\mathbf{Y}$ efficiently. Since the number of non-zero elements for row i in $\boldsymbol{\Sigma}^{tap}_{X_1}$ is determined by the number of measurement locations at a distance smaller than θ from location $\mathbf{s}_i$, the computational cost is determined both by the taper range and the spacing of the observations. Thus, if the measurements are irregularly spaced, it is difficult to get a precise estimate of the computational cost. However, for given measurement locations, the taper range can be chosen such that the average number of neighbors to the measurement locations is some fixed number k_θ. The cost for the Cholesky factorization is then similar to the cost for a GMRF with m nodes and a neighborhood size k_θ.

4.4. Covariance approximation

For practical applications of any of the approximation methods discussed here, one is often mostly interested in producing kriging predictions which are close to the optimal predictions. The error one makes in the kriging prediction is related to the method's ability to reproduce the true Matérn covariance function. There are many different wavelet bases one could consider using in the Markov approximation method, and we therefore compare some of these bases with respect to their ability to reproduce the Matérn covariance function in this section, so that we can choose only a few of the best bases to compare in the next section. As a reference, we also include the process convolution approximation in this comparison.

A natural measure of the error in the covariance approximation is a standardized L^2 norm of the difference between the true covariance function, $r(\mathbf{s}, \mathbf{u})$, and the covariance function for the approximation, $\hat{r}(\mathbf{s}, \mathbf{u})$,

$$\epsilon_r(\mathbf{s}) = \frac{\int (r(\mathbf{s},\mathbf{u}) - \hat{r}(\mathbf{s},\mathbf{u}))^2 \, \mathrm{d}\mathbf{u}}{\int r(\mathbf{s},\mathbf{u})^2 \, \mathrm{d}\mathbf{u}}. \tag{16}$$

Note here that $r(\mathbf{s}, \mathbf{u})$ is stationary and isotropic, while $\hat{r}(\mathbf{s}, \mathbf{u})$ generally is not. For the wavelet approximations and the process convolutions, ϵ_r is

periodic in $\mathbf{s}$ since the approximation error in general is smaller where the basis functions are centered, and we therefore use the mean value of $\epsilon_r(\mathbf{s})$ over the studied region as a measure of the covariance error.

We use the different methods to approximate the covariance function for a Matérn field on the square $[0, 10] \times [0, 10]$ in $\mathbb{R}^2$. The computational complexity for the kriging predictions depend on the number of basis functions, n, used in the approximations. For the Markov approximated spline bases and the Daubechies 3 basis, this complexity is $O(n^{3/2})$ whereas it is $O(n^3)$ for the spline bases if the Markov approximation is not used and for the process convolution method. We therefore use 100^2 basis functions for the $O(n^{3/2})$ methods and 100 basis functions for the other methods to get the covariance error for the methods when the computational cost is approximately equal.

Figure 3 shows the covariance error for the different methods as functions of the approximate range, $\kappa^{-1}\sqrt{8\nu}$, of the true covariance function for three different values of ν. There are several things to note in this figure:

1. The covariance error decreases for all methods as the range of the true covariance function increases. This is not surprising since the error will be small if the distance between the basis functions (which is kept fixed) is small compared to the true range.
2. The solid lines correspond to Markov approximations, which have computational complexity $\mathcal{O}(n^{3/2})$ for calculating the kriging predictor, and the approximations with computational complexity $\mathcal{O}(n^3)$ have dashed lines in the figure.
3. There is no convolution kernel estimate for $\nu = 1$ since the convolution kernel has a singularity at the origin in this case. For the other cases, the locations $\{u_j\}$ for the kernel basis functions were placed on a regular 10×10 lattice in the region.
4. The error one makes by the Markov approximation of the spline bases becomes larger for increasing order of the splines. Note that the third order spline basis is best without the approximation whereas the first order spline basis is best if the Markov approximation is used.

It is clear from the figure that the Markov approximations have lower covariance errors for the same computational complexity. Among these, the Daubechies 3 basis is best for large ranges whereas the Markov approximated first order spline basis is best for short ranges. The higher order spline bases have larger covariance errors so, from now on, we focus on the first order spline basis and the Daubechies 3 basis.

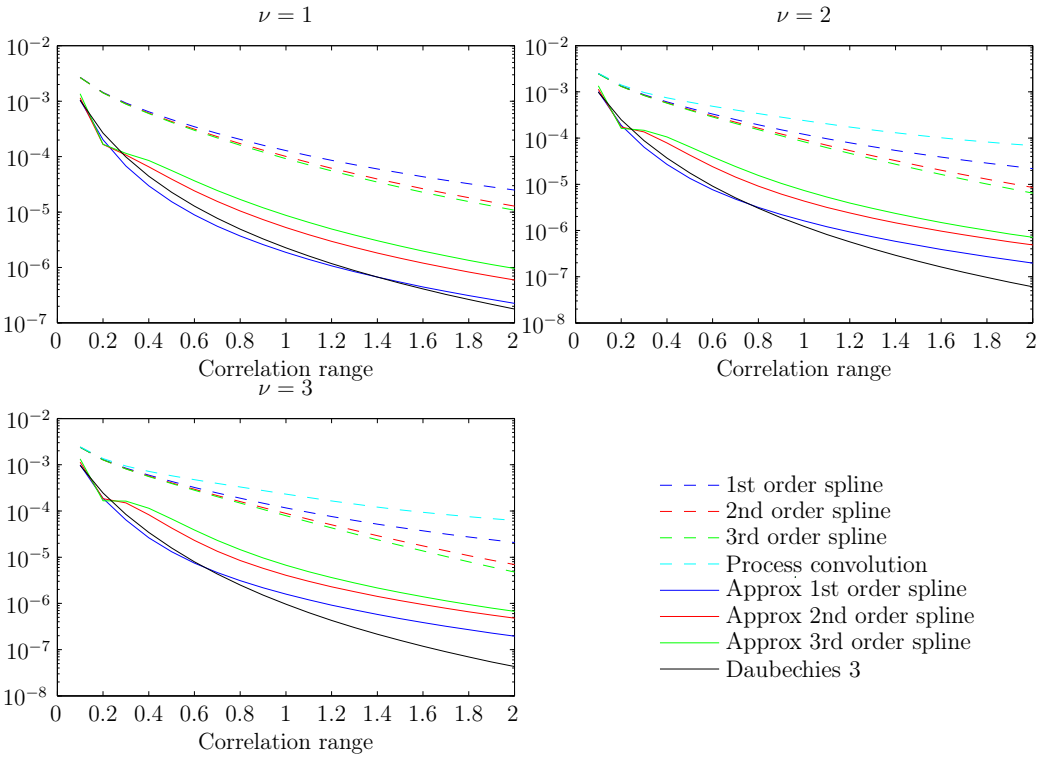

Figure 3: Numeric approximations of the L^2-norm (16) shown as a function of approximate range for different values of ν and different bases in $\mathbb{R}^2$. In this figure, 100^2 basis functions are used for the bases with Markov structure (solid lines), and 100 basis functions are used for the other bases (dashed lines). This gives approximately the same computational complexity for kriging prediction.

4.5. Spatial prediction

In the previous section, several bases were compared with respect to their ability to approximate the true covariance function when used in an approximation of the form (1) of a Gaussian Matérn field. The comparison showed that the Daubechies 3 (DB3) basis and the Markov approximated linear spline (S1) basis are most accurate for a given computational complexity. In this section, the spatial prediction errors for these two wavelet Markov approximations are compared with the process convolution method and the covariance tapering method. In the comparisons, note that the S1 basis is essentially of the same type of piecewise linear basis as used in Lindgren et al. (2011), so the results here also apply to that paper.

Simulation setup

Let $X(\mathbf{s})$ be a Matérn field with shape parameter ν (chosen later as 1, 2, or 3) and approximate correlation range r (later varied between 0.1 and 4). The range r determines κ through the relation $\kappa = \sqrt{8\nu}r^{-1}$ and the variance parameter $\phi = 4\pi\Gamma(\nu+1)\kappa^{\nu}\Gamma(\nu)^{-1}$ is chosen such that the variance of $X(\mathbf{s})$ is 1. We measure X at 5000 locations chosen at random from a uniform distribution on the square $[0,5]\times[0,5]$ in $\mathbb{R}^2$ using the measurement equation (2), where $\mathcal{E}(\mathbf{s})$ is Gaussian white noise uncorrelated with X with standard deviation $\sigma = 0.01$.

Given the measurements, spatial prediction of X to all locations on a 70×70 lattice of equally spaced points in the square is performed using the optimal kriging predictor, the wavelet Markov approximations, the process convolution method, and the covariance tapering method. For each approximate method, the sum of squared differences between the optimal kriging prediction and the approximate method's kriging prediction is used as a measure of kriging error.

We compare the methods for $\nu = 1, 2, 3$, and for each ν we test 40 different ranges varied between 0.1 and 4 in steps of 0.1. For a given ν and a given range, 20 data sets are simulated and the average kriging error is calculated for each method based on these data sets.

Choosing the number of basis functions

To obtain a fair comparison between the different methods, the number of basis functions for each method should be chosen such that the computation time for the kriging prediction is equal for the different methods. The computations needed for calculating the prediction can be divided into three main steps as follows

Step 1. Build all matrices except $\mathbf{M}$ in Step 3 necessary to calculate the kriging predictor.

Step 2. Solve the matrix inverse problem for the given method:

$$\begin{aligned}
&\text{S1, DB3 and Conv.:} && \mathbf{u} = (\boldsymbol{\Sigma}_w^{-1} + \mathbf{B}_1^{\top}\boldsymbol{\Sigma}_{\mathcal{E}}^{-1}\mathbf{B}_1)^{-1}\mathbf{B}_1\boldsymbol{\Sigma}_{\mathcal{E}}^{-1}\mathbf{Y},\\
&\text{Tapering:} && \mathbf{u} = \left(\boldsymbol{\Sigma}_{X_1}^{tap} + \boldsymbol{\Sigma}_{\mathcal{E}}\right)^{-1}\mathbf{Y},\\
&\text{Optimal kriging:} && \mathbf{u} = \left(\boldsymbol{\Sigma}_{X_1} + \boldsymbol{\Sigma}_{\mathcal{E}}\right)^{-1}\mathbf{Y}.
\end{aligned}$$

Step 3. Depending on which method that is used, build $\mathbf{M} = \mathbf{B}_2$, $\mathbf{M} = \boldsymbol{\Sigma}_{X_2X_1}^{tap}$, or $\mathbf{M} = \boldsymbol{\Sigma}_{X_2X_1}$ and calculate the kriging predictor $\hat{\mathbf{X}} = \mathbf{M}\mathbf{u}$.

For the optimal kriging predictor, and in some cases for the tapering method, the matrix $\mathbf{M}$ cannot be calculated and stored at once due to memory constraints if the number of measurements is large. Each element in $\hat{\mathbf{X}}$ is then constructed separately as $\hat{\mathbf{X}}_i = \mathbf{M}_i\mathbf{u}$, where $\mathbf{M}_i$ is a row in $\mathbf{M}$. It is then natural to include the time it takes to build the rows in $\mathbf{M}$ in the time it takes to calculate $\hat{\mathbf{X}}$, which is the reason for including the time it takes to build $\mathbf{M}$ in Step 3 instead of Step 1.

The computation time for the first step is highly dependent on the actual implementation, and we therefore focus on the computation time for the last two steps when choosing the number of basis functions. If kriging prediction is performed at a few locations only, the second step will dominate the computation time whereas the third step can dominate if kriging prediction is performed at many locations. To get results that are easier to interpret, we choose the number of basis functions such that the time for the matrix inverse problem in Step 2 is similar for the different methods.

Now since the computational complexity for Step 2 is $O(n^3)$ for the convolution method and $O(n^{3/2})$ for the Markov methods, one would think that if n basis functions are used in the convolution method and n^2 basis functions are used for the Markov methods, the computation time would be equal. Unfortunately it is not that simple. If two different methods have computational complexity $O(n^3)$, this means that the computation time scales as n^3 when n is increased for both methods; however, the actual computation time for a *fixed* n can be quite different for the two methods. For example, DB3 is approximately 6 times more computationally demanding than S1 for the same number of basis functions. The reason is that the DB3 basis functions have larger support than the S1 basis functions and this causes the matrices $\mathbf{B}_1$ and $\mathbf{\Sigma}_w^{-1}$ for DB3 to contain approximately 6 times as many nonzero elements compared to S1 for the same number of basis functions. However, the relative computation time will scale as $n_1^{3/2}$ if n_1 is increased for both methods.

To get approximately the same computation time for Step 2 for the different approximation methods, the number of basis functions for S1 is fixed to 100^2. Since DB3 is approximately six times more computationally demanding, the number of basis functions for this method is set to 1600. As mentioned in Lindgren et al. (2011), one should extend the area somewhat to avoid boundary effects from the SPDE formulation used in the Markov methods. We therefore expand the area with two times the range in each direction which results in a slightly higher number of basis functions used in

the computations.

The computation time for S1 and DB3 increases if ν is increased since the precision matrix for the weights contain more nonzero elements for larger values of ν. Therefore we use 625 basis functions placed on a regular 25×25 lattice in the kriging area for the convolution method when $\nu = 2$ and use 841 basis functions placed on a regular 29×29 lattice when $\nu = 3$. For the tapering method we chose the tapering range θ such that the expected number of measurements within a circle with radius θ to each kriging location is similar to the number of neighbors to the weights in the S1 method. For $\nu = 1$, $\nu = 2$, and $\nu = 3$ this gives a tapering ranges of 0.4, 0.55, and 0.7 respectively and results in approximately the same number of nonzero elements in the tapered covariance matrix as in the precision matrix for the S1 basis.

Results

Figure 4 shows the average kriging errors for the different methods as functions of the true covariance function's approximate range r. The values for given ν and r are averages of 20 simulations. The convolution kernels are singular if $\nu = 1$, so there is no convolution estimate for this case. The tapering estimate is best for short ranges, which is not surprising since the covariance matrix for the measurements is not much changed by the tapering if the true range then is shorter than the tapering range. For larger ranges, however, the tapering method has a larger error than the other methods. One reason for this is that the tapered covariance function is very different from the true covariance function if the true range is much larger than the tapering range. Another reason is that the prediction for all locations that do not have any measurements closer than the tapering range is zero in the tapering method. The convolution method has a similar problem if the effective range of the basis functions is smaller than the distance between the basis functions. In this case, the estimates for all locations that are not close to the center of some basis function have a large bias towards zero. These problems can clearly be seen in Figure 5, where the optimal kriging prediction, and the predictions for S1, the tapering method, and the convolution method, are shown for an example where $\nu = 2$ and the range is 1.

The computation times for the different methods are shown in Table 1. These computation times are obtained using an implementation in Matlab on a computer with a 3.33GHz Intel Xeon X5680 processor. As intended, the time for Step 2 is similar for the different methods whereas there is a larger

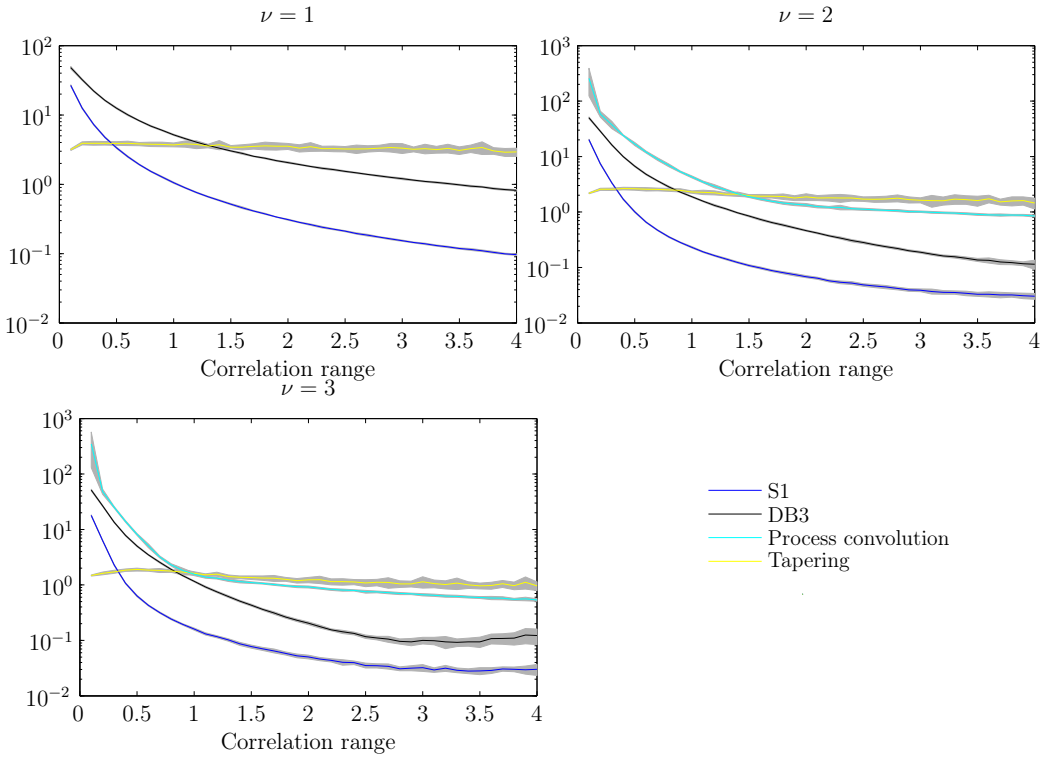

Figure 4: Kriging errors for the different methods as functions of the true covariance function's range. For each range, the values are calculated as the mean of 20 simulations. The lower limit of the bands around the curves is the estimate minus the standard deviation of the samples, and the upper limit is the estimate plus the standard deviation.

difference between the computation times for Step 3 because the computation time for the kriging prediction scales differently with the number of kriging locations for the different methods. Note that the wavelet methods are less computationally demanding than the tapering method and the convolution method when kriging prediction is performed at many locations. The reason being that the matrix $\mathbf{M}$ in Step 3 can be constructed without having to do costly covariance function evaluations.

As mentioned previously, the computation time for Step 1 is highly dependent on the actual implementation. However, as for Step 3, the Markov method's matrices can be constructed without performing any covariance function evaluations, which is the reason for the faster computation time. One thing to note here is that if the parameters are changed (for example during parameter estimation), one does not have to construct all matrices

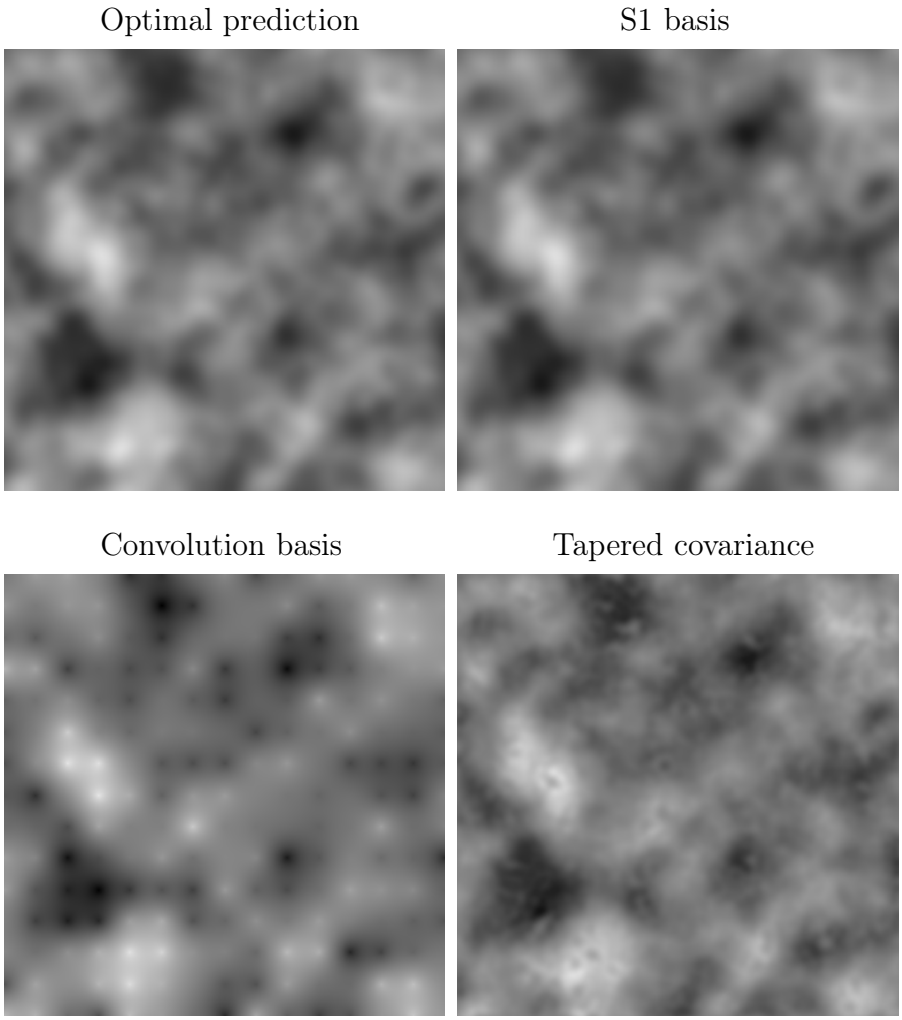

Figure 5: An example of an optimal kriging prediction and predictions using the S1 basis, the convolution basis, and a tapered covariance when $\nu = 2$ and the covariance range is 1. The predictions are based on 5000 observations and are calculated for a 200×200 grid in the square $[0, 5] \times [0, 5]$. The number of basis functions and the tapering range are chosen such that the total time for Step 2 and Step 3 is approximately equal for the different methods.

$\nu = 1$

	Step 1	Step 2	Step 3	Total
Optimal	37.68 (6.357)	5.074 (0.277)	36.48 (6.231)	79.23 (8.906)
DB3	0.490 (0.049)	0.113 (0.014)	0.293 (0.026)	0.896 (0.057)
S1	0.423 (0.027)	0.088 (0.007)	0.248 (0.018)	0.759 (0.033)
Conv.	– –	– –	– –	– –
Taper	2.771 (0.191)	0.117 (0.010)	2.051 (0.127)	4.939 (0.229)

$\nu = 2$

	Step 1	Step 2	Step 3	Total
Optimal	36.19 (6.965)	5.327 (0.529)	34.94 (6.695)	76.45 (9.675)
DB3	0.600 (0.090)	0.228 (0.039)	0.310 (0.049)	1.138 (0.110)
S1	0.489 (0.055)	0.203 (0.025)	0.260 (0.036)	0.951 (0.070)
Conv.	0.961 (0.027)	0.217 (0.019)	0.942 (0.027)	2.120 (0.043)
Taper	4.184 (1.523)	0.247 (0.028)	3.319 (0.251)	7.750 (1.543)

$\nu = 3$

	Step 1	Step 2	Step 3	Total
Optimal	42.75 (6.572)	5.468 (0.380)	41.36 (6.440)	89.58 (9.210)
DB3	0.759 (0.091)	0.394 (0.051)	0.315 (0.033)	1.468 (0.110)
S1	0.569 (0.042)	0.377 (0.035)	0.266 (0.025)	1.213 (0.060)
Conv.	5.656 (1.094)	0.390 (0.024)	5.522 (1.078)	11.57 (1.537)
Taper	6.413 (1.051)	0.421 (0.035)	5.460 (0.402)	12.30 (1.126)

Table 1: Average computation times in seconds for the results in Figure 4. The values are based on the 800 simulations for each value of ν. The standard deviations are shown in the parentheses.

again in the Markov methods as one has to do for the other two methods.

In conclusion we see that S1 is both faster and has a smaller kriging error for all ranges when compared to DB3 and the convolution method and compared to the tapering method it has a smaller kriging error for all but very short ranges. Since the tapering method's computational cost varies with the tapering range, we conclude this section with a study of how changing the tapering range changes the results in order to get a better understanding of which method is to prefer when comparing S1 and the tapering method.

4.6. A study of varying the tapering range

As shown above, the S1 method should be preferred over the DB3 method and the convolution method in all our test cases whereas the tapering method had a smaller kriging error for very short ranges. Since this was done using a fixed tapering range, chosen such that the computation time for Step 2 would be similar to the other methods, we now look at what happens if the tapering range is varied when keeping the true range fixed.

The setup is the same as in the previous comparison, a Matérn field with $\nu = 2$, variance 1, and an approximate range r is measured at 5000 randomly chosen locations in a square in $\mathbb{R}^2$. The difference is that we now keep these parameters fixed but instead vary the tapering range from 0.05 to 2 in steps of 0.05. We generate 100 data sets and calculate the kriging predictions for the S1 method and the tapering method for all values of the tapering range. Based on these 100 estimates, the average kriging error is calculated for S1 and for each tapering estimate.

The results can be seen in Figure 6. The kriging errors are shown in the left panels and the computation times are shown in the right panels. The blue lines represent the S1 method, which obviously does not depend on the tapering range, and the yellow lines represent the tapering method. In the left panels, the solid lines show the time for Step 2 in the computations and the dashed lines show the total time for Step 2 and Step 3. In the upper two panels, the true range r is 1, and 100^2 S1 basis functions are used. In this case, S1 is more accurate than the tapering method for all tapering ranges tested, which is not surprising considering the previous results. In the bottom panels of the figure, the true range r is 0.25 and 100^2 S1 basis functions are used. This is a situation where the tapering method was more accurate than S1 in the previous study and we see here that the tapering method is more accurate for tapering ranges larger than 0.4 and that the time for Step 2 is smaller for all tapering ranges smaller than 0.46. Thus, by choosing the tapering range between 0.4 and 0.46, the tapering method is more accurate and has a smaller computation time for Step 2.

The accuracy of the tapering method increases if the ratio between the tapering range and the true range is increased, and the computation time depends on what the distance between the measurements is compared to the tapering range. If the distance between the measurements is large, the tapering method is fast, whereas it is slower if the distance is small. Thus, the situation where the tapering method performs best is when the true covariance range is short compared to the distance between the measurements.

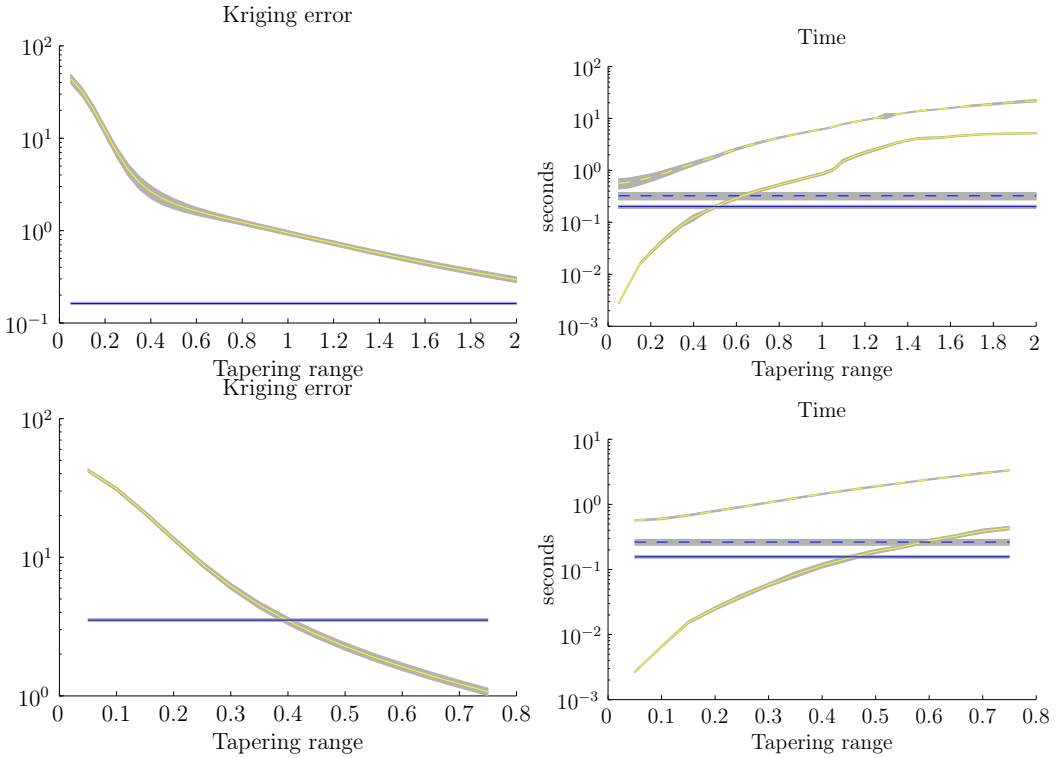

Figure 6: The computation times (right) and kriging errors (left) for the covariance tapering method (yellow lines) as functions of the taper range. Values for the S1 basis (blue lines) are shown for comparison. The range of the true covariance function is 1 (upper panels) and 0.25 (lower panels). The results are averages of 100 simulations, and the grey bands indicate the standard deviation of these samples. The solid lines in the right panels show the computation time for Step 2 and the dashed lines show the total computation time for Step 2 and Step 3.

However, also for the case when the true range is small, the total time it takes to calculate the tapering prediction is larger than the time it takes to calculate the S1 prediction unless the number of kriging locations is small.

In this work, the taper functions that Furrer et al. (2006) found to be best for each value of ν are used, but the results may be improved by using other taper functions. Changing the taper function will, however, not change the fact that the prediction for all locations that do not have any measurements closer than the tapering range is zero in the tapering method and that the tapered covariance function is very different from the true covariance function if the tapering range is short compared to the true range. Finally, the results

for all methods could be improved by finding optimal parameters for the approximate models instead of using the parameters for the true Matérn covariance. For the tapering method, however, Furrer et al. (2006) found that this only changed the relative accuracy by one or two percent.

5. Conclusions

Because of the increasing number of large environmental data sets, there is a need for computationally efficient statistical models. To be useful for a broad range of practical applications, the models should contain a wide family of stationary covariance functions, and be extendable to nonstationary covariance structures, while still allowing efficient calculations for large problems.

The SPDE formulation of the Matérn family of covariance functions has these properties, as it can be extended to more general nonstationary spatial models (see Bolin and Lindgren, 2011; Lindgren et al., 2011, for details on how this can be done), and allows for efficient and accurate Markov model representations. In addition, as shown by the simulation comparisons, these Markov methods are more efficient and accurate than both the process convolution approach and the covariance tapering method for approximating stationary and isotropic Matérn fields with $\nu + d/2 \in \mathbb{N}$.

Depending on the context in which a model is used, different aspects are important to make it computationally efficient. If, for example, the model is used in MCMC simulations, one should be able to generate samples from the model given the parameters efficiently, or if the parameters are estimated in a numerical maximum likelihood procedure, one must be able to evaluate the likelihood efficiently. To limit the scope of this article, only the computational aspects of kriging was considered. However, for practical applications, parameter estimation is likely the most computationally demanding part of the analysis. If maximum likelihood estimation is performed using numerical optimization of the likelihood, matrix inverses similar to the one in Step 2 in Table 1 have to be performed in each iteration of the optimization, and it is therefore important that these inverses can be calculated efficiently. We have not discussed estimation here, but the Markov methods are likely most efficient in this situation as well because these do not require costly Bessel function evaluations when calculating the likelihood. However, this is left for future research to investigate in more detail. An introduction to maximum

likelihood estimation using the SPDE formulation can be found in Bolin and Lindgren (2011) and Lindgren et al. (2011).

Finally, some relevant methods, such as fixed rank kriging (Cressie and Johannesson, 2008) and predictive process models (Banerjee et al., 2008; Eidsvik et al., 2012), were not included in the comparison in order to keep it relatively short and also because they are difficult to compare with the methods discussed here. Also, a method that likely would give better results than the tapering method is to use the compactly supported covariance functions by Gneiting (1999) adapted to the Matérn covariance function. How to adapt these compactly supported covariance functions optimally is, however, not clear, and we therefore leave this for future work as well.

Acknowledgements

We are thankful for the comments by the reviewers and the editor which led to a greatly improved manuscript.

References

Banerjee, S., Gelfand, A.E., Finley, A.O., Sang, H., 2008. Gaussian predictive process models for large spatial data sets. J. Roy. Statist. Soc. Ser. B Stat. Methodol. 70, 825–848.

Barry, R.P., Ver Hoef, J.M., 1996. Blackbox kriging: Spatial prediction without specifying variogram models. J. Agr. Biol. Environ. Statist. 1, 297–322.

Bolin, D., Lindgren, F., 2011. Spatial models generated by nested stochastic partial differential equations, with an application to global ozone mapping. Ann. Appl. Statist. 5, 523–550.

Burrus, C., Gopinath, R., Guo, H., 1988. Introduction to Wavelets and Wavelet Transforms: A Primer. Prentice-Hall, New York.

Chiles, J.P., Delfiner, P., 1999. Geostatistics, Modeling Spatial uncertainty. Wiley Series in Probability and statistics.

Chui, C.K., Wang, J.Z., 1992. On compactly supported spline wavelets and a duality principle. T. Am. Math. Soc. 330, 903–915.

Cressie, N., Johannesson, G., 2008. Fixed rank kriging for very large spatial data sets. J. Roy. Statist. Soc. Ser. B Stat. Methodol. 70, 209–226.

Cressie, N., Ravlicová, M., 2002. Calibrated spatial moving average simulations. Statist. Model. 2, 267–279.

Daubechies, I., 1992. Ten Lectures on Wavelets (CBMS-NSF Regional Conference Series in Applied Mathematics). Soc for Industrial & Applied Math.

Eidsvik, J., Finley, A.O., Banerjee, S., Rue, H., 2012. Approximate bayesian inference for large spatial datasets using predictive process models. Computational Statistics and Data Analysis 56, 1362 – 1380.

Furrer, R., Genton, M.G., Nychka, D., 2006. Covariance tapering for interpolation of large spatial datasets. J. Comput. Graph. Statist. 15, 502–523.

Gelfand, A., Diggle, P., Guttorp, P., 2010. Handbook of spatial statistics. Chapman & Hall/CRC handbooks of modern statistical methods, Taylor & Francis Group.

Gneiting, T., 1999. Correlation functions for atmospheric data analysis. Q. J. R. Meterol. Soc. 125, 2449–2464.

Gneiting, T., 2002. Compactly supported correlation functions. J. Multivariate Anal. 83, 493–508.

Higdon, D., 2001. Space and Space-time modeling using process convolutions. Technical Report 01-03. Duke University, Durham, NC.

Latto, A., Resnikoff, H.L., Tenenbaum, E., 1991. The evaluation of connection coefficients of compactly supported wavelets, in: Proceedings of the French-USA Workshop on Wavelets and Turbulence, Springer-Verlag.

Lindgren, F., Rue, H., 2007. Explicit construction of GMRF approximations to generalised Matérn fields on irregular grids. Preprints in Math. Sci. Lund University 2007:12.

Lindgren, F., Rue, H., Lindström, J., 2011. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach (with discussion). J. Roy. Statist. Soc. Ser. B Stat. Methodol. 73, 423–498.

Matérn, B., 1960. Spatial variation. Meddelanden från statens skogsforskningsinstitut 49.

Nychka, D., Wikle, C., Royle, J.A., 2002. Multiresolution models for nonstationary spatial covariance functions. Statist. Model. 2, 315–331.

Rodrigues, A., Diggle, P.J., 2010. A class of convolution-based models for spatio-temporal processes with non-separable covariance structure. Scand. J. Statist. 37, 553–567.

Rue, H., Held, L., 2005. Gaussian Markov Random Fields; Theory and Applications. volume 104 of *Monographs on Statistics and Applied Probability*. Chapman & Hall/CRC.

Rue, H., Tjelmeland, H., 2002. Fitting Gaussian Markov random fields to Gaussian fields. Scand. J. Statist. 29, 31–49.

Schabenberger, O., Gotway, C., 2005. Statistical methods for spatial data analysis. Texts in statistical science, Chapman & Hall/CRC.

Song, H., Fuentes, M., Gosh, S., 2008. A comparative study of Gaussian geostatistical models and Gaussian Markov random field models. J. Multivariate Anal. 99, 1681–1697.

Stein, M.L., 1999. Interpolation of Spatial Data: Some Theory for Kriging. Springer-Verlag, New York.

Wendland, H., 1995. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. Adv. Comput. Math. 4, 389–396.

Whittle, P., 1963. Stochastic processes in several dimensions. Bull. Internat. Statist. Inst. 40, 974–994.

Zhang, H., 2004. Inconsistent estimation and asymptotically equal interpolations in model-based geostatistics. J. Amer. Statist. Assoc. 99, 250–261.